

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 08069403 A

(43) Date of publication of application: 12 . 03 . 96

(51) Int. Cl

G06F 12/00
G06F 12/00

(21) Application number: 06223984

(71) Applicant: FUJI XEROX CO LTD

(22) Date of filing: 26 . 08 . 94

(72) Inventor: SANO MAKOTO

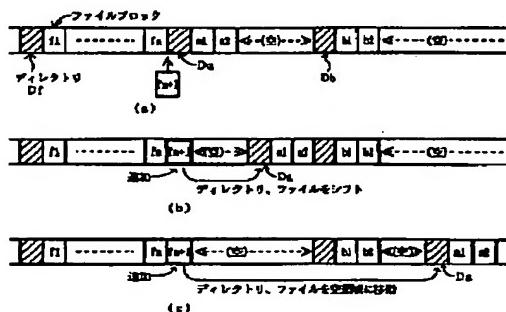
(54) FILE MANAGEMENT DEVICE

(57) Abstract:

PURPOSE: To provide the file managing device which is short in access time.

CONSTITUTION: The file managing device which stores a file in a specific continuous area secured on an external storage device has a moving means which shifts or moves adjacent data or all of its data to another area unless the secured area has a vacancy when the file is expected to be expanded and a means which expands the file by using the continuous free area formed by the movement. The specific continuous area is secured in directory units and the moving means is a directory moving means and a directory expanding means which incorporates the continuous free area formed by the movement in a desired directory.

COPYRIGHT: (C)1996,JPO



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-69403

(43)公開日 平成8年(1996)3月12日

(51)Int.Cl.⁶
G 0 6 F 12/00

識別記号 庁内整理番号
5 2 0 P 7623-5B
5 0 1 M 7623-5B

F I

技術表示箇所

審査請求 未請求 請求項の数4 FD (全7頁)

(21)出願番号 特願平6-223984

(22)出願日 平成6年(1994)8月26日

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂三丁目3番5号

(72)発明者 佐野 誠

神奈川県厚木市中町4丁目9番17号原田ビル

富士ゼロックス株式会社内

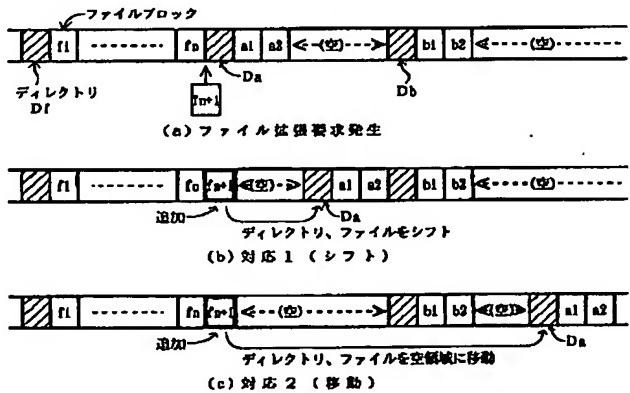
(74)代理人 弁理士 平木 道人 (外1名)

(54)【発明の名称】 ファイル管理装置

(57)【要約】

【目的】 アクセス時間の短いファイル管理装置を提供すること。

【構成】 外部記憶装置に確保された所定の連続領域にファイルを格納していくファイル管理装置において、ファイルを拡張しようとした時に、確保された連続領域に空きが無い場合には、隣接するデータあるいは自分自身のデータの内のいずれかの全データをシフトするかあるいは他の領域に移動させる移動手段と、移動により生じた連続する空き領域を使用してファイルを拡張する手段とを有する。また所定の連続領域はディレクトリ単位に確保され、前記移動手段は、ディレクトリ移動手段であり、かつ、移動により生じた連続する空き領域を所望のディレクトリに組み込むディレクトリ拡張手段を有してもよい。



【特許請求の範囲】

【請求項 1】 外部記憶装置に確保された所定の連続領域にファイルを格納していくファイル管理装置において、ファイルを拡張しようとした時に、確保された連続領域に空きが無い場合には、隣接するデータあるいは自分自身のデータの内のいずれかの全データを移動させる移動手段と、移動により生じた連続する空き領域を使用してファイルを拡張する手段とを有することを特徴とするファイル管理装置。

【請求項 2】 前記所定の連続領域はディレクトリ単位に確保され、前記移動手段は、ディレクトリ内の全データを移動させるディレクトリ移動手段であり、かつ、移動により生じた連続する空き領域を拡張すべきファイルが含まれるディレクトリに組み込むディレクトリ拡張手段を有することを特徴とする請求項 1 に記載のファイル管理装置。

【請求項 3】 前記ディレクトリ移動手段は、隣接するディレクトリのアクセス頻度チェック手段と、隣接するディレクトリの空き領域チェック手段と、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、移動するディレクトリを決定する移動ディレクトリ決定手段を含むことを特徴とする請求項 2 に記載のファイル管理装置。

【請求項 4】 前記移動手段は、隣接するディレクトリのアクセス頻度チェック手段と、隣接するディレクトリの空き領域チェック手段と、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、ディレクトリをシフトするかあるいは離れた空き領域に移動するかの移動形式を決定する移動形式決定手段とを含むことを特徴とする請求項 2 に記載のファイル管理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はコンピュータ等におけるファイル管理装置に関し、特にファイル管理装置における外部記憶装置へのページの割り当て制御方式に関するものである。

【0002】

【従来の技術】 ハードディスク、フロッピディスク、光ディスク等の外部記憶装置を有するコンピュータにおいては、ファイルをページ（外部記憶装置のフォーマットに基づく所定の長さのデータ）単位で記憶し、管理している。例えばUNIXにおける外部記憶装置、例えばハードディスク内のデータ構造は、図3に示すように、大きく4つのブロックに分かれ。まず、ブートブロックには起動時に読み込まれるプログラムが格納されている。ブロック管理領域（スーパーブロック）には各領域の大きさ、空きページの管理のための情報等が格納されている。iノード領域（iノードリスト）には複数のiノードと呼ばれるテーブルが格納されている。iノード

には、ファイルの種類、大きさ、ファイルの実際のデータが格納されている複数のデータブロックへのアドレス（ページアドレス）等の情報が格納されており、1つのiノードが1つのファイルに対応する。

【0003】 データ領域には実際のファイルデータおよびディレクトリがページ単位で分散して格納されている。ディレクトリには、該ディレクトリ内の複数のファイル名およびファイル名に対応するiノード番号とを記憶したテーブルが格納されている。ファイルにアクセスする場合には、まずディレクトリ内の目的とするファイル名を検索し、対応するiノード番号を求める。つぎにiノード領域中の目的のiノードを読み出し、iノード中のデータブロックへのアドレス（ページアドレス）を基にデータ領域からファイルデータを読み出す。

【0004】 このような制御方式においては、データ領域に離散的に空き領域が存在する場合でも、1つのファイルのデータをそれぞれの空き領域に割り当てる、各領域を1つのiノードに登録することによりファイルを格納することが可能となる。ところが、ファイルの生成、削除を繰り返していくと、連続領域にファイルを割り当てることができなくなり、ファイルの記憶位置の断片化が進行する。記憶位置が断片化すると、点在しているページを読み込むために、ヘッドの移動時間やディスクの回転待ち時間が多く必要となる。従って、従来の制御方式においては、アクセス時間が遅くなってしまうという問題点があった。このような問題点を解決する方法として、例えば、特開平2-19938号公報には、初期化時に複数の異なるサイズの連続空き領域を設けて、ファイルを割り当てる方式が記載されている。

【0005】 【発明が解決しようとする課題】 上記公報に記載されているような従来のファイル管理装置においては、連続する空き領域を全て使用してしまった場合には、後続するデータが離れたアドレスに割り当たる可能性があり、従ってアクセス時間が長くなるという問題点があった。本発明の目的は、前記のような従来技術の問題点を解決し、アクセス時間の短いファイル管理装置を提供することにある。

【0006】

【課題を解決するための手段】 第1の発明は、外部記憶装置に確保された所定の連続領域にファイルを格納していくファイル管理装置において、ファイルを拡張しようとした時に、確保された連続領域に空きが無い場合には、隣接するデータあるいは自分自身のデータの内のいずれかの全データを移動させる移動手段と、移動により生じた連続する空き領域を使用してファイルを拡張する手段とを有することを特徴とする。第2の発明は、第1の発明において、所定の連続領域はディレクトリ単位に確保され、移動手段はディレクトリ内の全データを移動させるディレクトリ移動手段であり、かつ、移動により

生じた連続する空き領域を拡張すべきファイルが含まれるディレクトリに組み込むディレクトリ拡張手段を有することを特徴とする。

【0007】第3の発明は、第2の発明において、移動手段は、隣接するディレクトリのアクセス頻度チェック手段と、隣接するディレクトリの空き領域チェック手段と、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、移動するディレクトリを決定するディレクトリ決定手段を含むことを特徴とする。第4の発明は、第2の発明において、移動手段は、隣接するディレクトリのアクセス頻度チェック手段と、隣接するディレクトリの空き領域チェック手段と、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、ディレクトリをシフトするかあるいは離れた空き領域に移動するかの移動形式を決定する移動形式決定手段とを含むことを特徴とする。

【0008】

【作用】第1の発明においては、ファイルを拡張しようとした場合に、もし連続領域に空きがなければ、前後あるいは自分自身のデータをシフトまたは他の空き領域に移動させることにより、連続領域を拡張する。従って、ファイルには常に連続する領域が割り当てられているので、アクセスが高速になり、スループットが向上する。第2の発明においては、あるディレクトリにファイルを追加するかあるいはファイルを拡張しようとした場合に、もし該ディレクトリに割り当てられた連続領域に空きがなければ、前後あるいは自分自身のディレクトリをシフトまたは他の空き領域に移動させることにより、該ディレクトリの連続領域を拡張する。従って、該ディレクトリには常に連続する領域が割り当てられているので、アクセスが高速になり、スループットが向上する。

【0009】第3の発明においては、第2の発明において、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、移動するディレクトリを決定するディレクトリ決定手段を備えたことにより、最も効率のよいディレクトリ移動処理を選択可能となり、処理時間が短縮される。第4の発明においては、第2の発明において、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、ディレクトリをシフトするかあるいは離れた空き領域に移動するかの移動形式を決定する移動形式決定手段を備えたことにより、やはり最も効率のよいディレクトリ移動処理を選択可能となり、処理時間が短縮される。

【0010】

【実施例】図2は、本発明が適用されるコンピュータシステムの一例を示すブロック図である。CPU1は、RAM2内に記憶されるプログラムに基づき各種の処理を行う中央処理装置である。RAM2は、ディスクから読

み出された、CPU1が動作するためのプログラムを記憶する他、各種のバッファやワークエリアとして使用される。コンソールインターフェース3は、表示信号をCRTや液晶を用いたディスプレイ4に出力し、またキーボード5から入力されたコード情報をCPU1に通知する。ハードディスクコントローラHDC6は、CPU1からの制御に基づき、プログラムあるいはデータのファイルを格納しているハードディスクHD7の読み出しあるいは書き込み制御を行う。I/Oインターフェース8は、フロッピィディスク、光ディスク等の各種記憶装置、あるいはスキャナ、プリンタ、LAN等の各種入出力装置であるI/O9との接続、制御を行う。バス10はシステム内の各回路を接続している。

10

【0011】図4は、本発明において、ディレクトリ毎にディレクトリに後続する所定の連続領域を割り当てる場合のファイルデータの格納状態を示す説明図である。

図4においては、ディレクトリに2つのファイルa、bが登録されており、ファイルa、bはそれぞれiノード番号100および200に対応している。そして、iノード100および200には、ファイルaの3つのデータブロック(ページ)a1、a2、a3に対応するページアドレスおよびファイルbの2つのデータブロックb1、b2に対応するページアドレスがそれぞれ記憶されている。後続のディレクトリとの間には未使用の空きブロック(506～508)が存在する。

20

【0012】図5は、本発明におけるディレクトリの詳細な内容を示す説明図である。ディレクトリ内には、ファイル名とiノード番号との対応テーブルの他に、ディレクトリ内のブロック数(図4の例では8)、ディレク

30

トリ内の空きブロックリスト、ディレクトリのアクセス頻度N、ディレクトリ移動要求フラグの情報が記憶されている。ディレクトリのアクセス頻度Nは、該ディレクトリ内の任意のファイルに対する単位時間(例えば1秒)当たりのアクセス回数の情報であり、ディレクトリ内にある図示しないカウンタ等によって計測される。ディレクトリ移動要求フラグは、ディレクトリの拡張要求が発生した時に、ディレクトリのシフトや移動によってディレクトリの拡張ができない場合にセットされるフラグであり、ディレクトリやファイルの削除、移動等によって新たな空き領域が発生した場合に、該フラグのセットされているディレクトリの拡張が試みられる。

40

【0013】図6は、本発明におけるディレクトリの生成処理を示すフローチャートである。ステップS10においては、ディレクトリの生成に必要な連続した所定ブロックからなる空き領域があるか否かが判定され、結果が否定であればステップS12に移行するが、肯定であればステップS11に移行する。ステップS11においてはトラック上の前のディレクトリに対して所定の領域を空けてディレクトリを生成する。ステップS13においては、生成した図5のディレクトリに、ディレクトリ

5
内のブロック数、ディレクトリ内の空きブロックリストを登録する。十分な空き領域が確保出来ない場合には、ステップS12において確保出来る範囲で最大の領域を選択してディレクトリを生成する。

【0014】図1は、前記したような方式で、各ディレクトリにトラック上で後続する所定の連続領域を割り当てた場合に、該領域に空きが無くなつた場合のディレクトリ拡張処理を示す説明図である。各ディレクトリDf、Da、Dbはそれぞれトラック上で等間隔に配置されており、ディレクトリDbの後ろは空いているものとする。今、図1(a)に示すように、ディレクトリDfには空き領域が無く、更にファイルブロックfn+1の追加が要求されたものとする。

【0015】この場合の対応策としては、図1(b)に示すように、後続するディレクトリDaに空き領域が十分あれば、ディレクトリDaおよび該ディレクトリ内のファイルブロック(a1、a2)を後ろにシフトさせて、ファイルブロックfnの後ろに空き領域を生成する。そして追加すべきファイルブロックfn+1をfnの後ろに追加する。シフトする量は、(b)のようにディレクトリDaの空き領域が無くなるまでシフトしてもよいし、またディレクトリDaとDfの空き領域が等しくなるようにするか、あるいはファイルブロックの追加に必要な最小限のシフトを行うようにしてもよい。図1(c)は第2の対応策を示しており、後続するディレクトリDaを他の空き領域に全て移動させ、空き領域を生成する。この場合、空き領域の前のディレクトリとは所定の領域を空けて移動させ、該ディレクトリの領域を確保するようにする。実際には、後述するように、隣接ブロックの空きの有無、アクセス頻度、他の空き領域の有無等を考慮して最も処理の速い対応策を採用する。

【0016】図7はディレクトリ内にファイルブロックを割り当てる処理を示すフローチャートである。ステップS20においては、図5の空きブロックリストを参照することによって、ディレクトリにファイルを格納できるだけの空き領域があるか否かが判定され、結果が肯定であればステップS33に移行するが、否定の場合にはステップS21に移行する。ステップS21においては、トラック上の後ろのディレクトリのアクセス頻度が平均値よりも高いか否かが判定される。アクセス頻度の平均値は、別途カウントした単位時間当たりのファイルへの総アクセス回数をディレクトリ数で除算することにより算出される。なお、平均値の代わりに領域を拡張すべきディレクトリのアクセス頻度と比較してもよい。また、アクセス頻度を考慮する理由は、ディレクトリの移動中はディレクトリ内のファイルへのアクセスが禁止されるので、アクセス頻度の高いディレクトリを移動すると、アクセス待ちが多発してスループットが低下する可能性が高いためである。

【0017】ステップS21の判定結果が否定の場合に

はステップS22に移行し、空きブロックリストを参照することによって、後ろのディレクトリの空き領域が十分あるか否かが判定され、結果が肯定であればステップS23に移行して、図1(b)に示すように、後ろのディレクトリ全体を後ろにシフトして領域を縮小させる。また空き領域が十分なければステップS24に移行し、他に後ろのディレクトリを移動できる空き領域があるか否かが判定され、結果が否定であればステップS26に移行するが、肯定の場合にはステップS25に移行して、図1(c)に示すように、後ろのディレクトリを他の空き領域に移動させる。

【0018】ステップS26においては、トラック上の前のディレクトリのアクセス頻度が平均値（あるいは自分のアクセス頻度）よりも高いか否かが判定される。ステップS26の判定結果が否定の場合にはステップS27に移行し、前のディレクトリの空き領域が十分あるか否かが判定され、結果が肯定であればステップS30に移行して、自分自身のディレクトリ全体を前にシフトして領域を拡張させる。また空き領域が十分なければステップS28に移行し、他に前のディレクトリを移動できる空き領域があるか否かが判定され、結果が否定であればステップS34に移行するが、肯定の場合にはステップS29に移行して、前のディレクトリを他の空き領域に移動させ、ステップS30に移行して自分自身を前にシフトさせる。

【0019】ステップS26における判定結果が肯定の場合にはステップS31に移行し、自分自身のディレクトリを移動できる空き領域があるか否かが判定され、結果が肯定の場合にはステップS32に移行し、自分自身のディレクトリを他の空き領域に移動させる。そしてステップS33においては、ディレクトリに割り当てられた領域にファイルブロックを追加し、ディレクトリおよび対応するiノードの情報を更新する。

【0020】ステップS31の判定結果が否定の場合にはステップS34に移行し、ディレクトリ内のディレクトリ移動要求フラグをオンにセットし、ステップS35においては、とりあえず他の空き領域にファイルブロックを格納し、ディレクトリおよび対応するiノードの情報を更新する。オペレーティングシステムのファイル管理システムにおいては、ファイルやディレクトリの削除、移動等によりディスクに空き領域が生成された場合には、ディレクトリ移動要求フラグの立っているディレクトリを移動可能か否かが判定され、結果が肯定であれば移動される。以上のような処理により、ディレクトリに割り当てられた領域が不足した場合には、なるべく連続する領域を拡張して割り当てるようにしたので、アクセス時間が速くなり、スループットが向上する。

【0021】以上、実施例を説明したが、本発明は以下に示すような変形例も考えられる。実施例としては、ディレクトリ単位でシフトあるいは移動を行う例を開示し

たが、ディレクトリの代わりにファイル単位で所定の連続領域を割り当てるようなシステムにおいても本発明を適用可能である。図7の移動ディレクトリ及び移動方法の決定処理は一例であって、例えばステップS25における後ろのディレクトリの移動よりもステップS30の自分自身のディレクトリのシフトの方が処理が速い場合には自分自身のシフトを選択するようにしてもよい。即ち、前後と自分自身のディレクトリのデータ量(ページ数)および空き容量、アクセス頻度、位置、ディスクの特性等を考慮して最も短時間で実行可能な処理を選択するようにすればよい。更に、2つ以上離れたディレクトリの空き容量まで考慮して、複数のディレクトリのシフトあるいは移動とシフトを行うようにしてもよい。本発明はハードディスクに適用する例を開示したが、フロッピディスク、光ディスク等任意の動的記憶装置あるいは磁気バブル、シフトレジスタ等のシーケンシャルな静的記憶装置に適用可能である。

【0022】

【発明の効果】第1の発明においては、ファイルを拡張しようとした場合に、もし連続領域に空きがなければ、前後あるいは自分自身のファイルデータをシフトするかあるいは他の空き領域に移動させることにより、連続領域を拡張する。従って、ファイルには常に連続する領域が割り当てられるので、アクセスが高速になり、スループットが向上する。第2の発明においては、あるディレクトリにファイルを追加するかあるいはファイルを拡張しようとした場合に、もし該ディレクトリに割り当てられた連続領域に空きがなければ、前後あるいは自分自身のディレクトリをシフトまたは他の空き領域に移動させることにより、該ディレクトリの連続領域を拡張する。従って、該ディレクトリには常に連続する領域が割り当てられているので、やはりアクセスが高速になり、スループットが向上するという効果がある。*

* 【0023】第3の発明においては、第2の発明において、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、移動するディレクトリを決定するディレクトリ決定手段を備えたことにより、最も効率のよいディレクトリ移動処理を選択可能となり、処理時間が短縮される。第4の発明においては、第2の発明において、アクセス頻度チェック手段および空き領域チェック手段の少なくとも一方の出力に基づき、ディレクトリをシフトするかあるいは離れた空き領域に移動するかの移動形式を決定する移動形式決定手段を備えたことにより、やはり最も効率のよいディレクトリ移動処理を選択可能となり、処理時間が短縮されるという効果がある。

【図面の簡単な説明】

【図1】本発明におけるディレクトリ拡張方式を示す説明図である。

【図2】本発明が適用される計算機の一例を示すプロック図である。

【図3】ハードディスク内のデータ構造を示す説明図である。

【図4】ディレクトリ毎に所定の連続領域を割り当てた場合のファイルデータの格納状態を示す説明図である。

【図5】本発明におけるディレクトリの詳細な内容を示す説明図である。

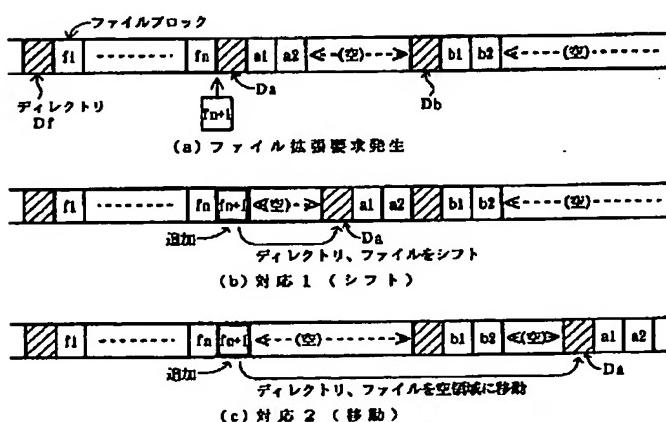
【図6】本発明のディレクトリ生成処理を示すフローチャートである。

【図7】本発明のファイブロック割当処理を示すフローチャートである。

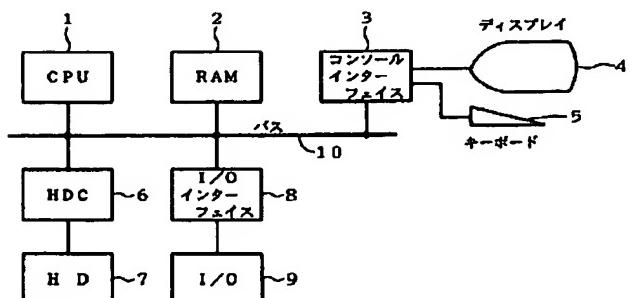
【符号の説明】

30 1…CPU、2…RAM、3…コンソールインターフェース、4…ディスプレイ、5…キーボード、6…ハードディスクコントローラ、7…ハードディスク、8…I/Oインターフェース、9…入出力機器、10…バス

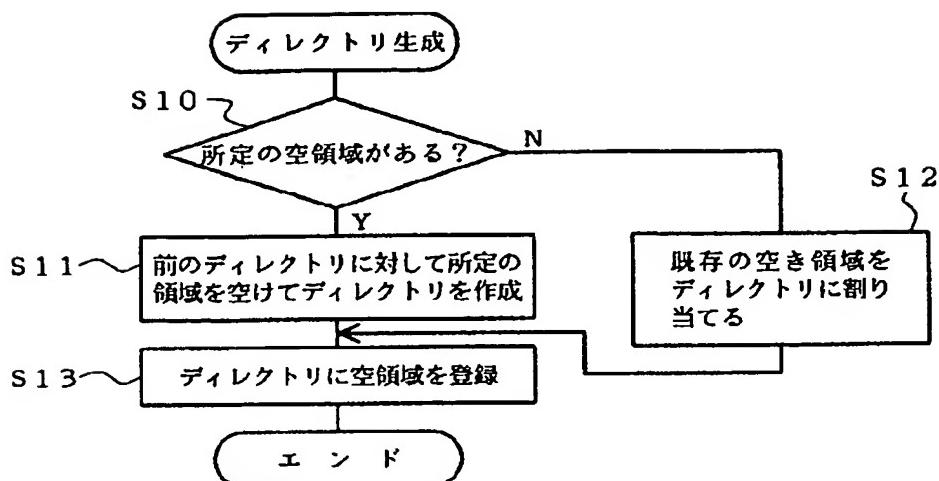
【図1】



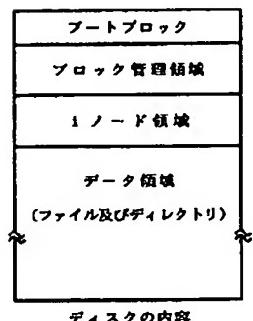
【図2】



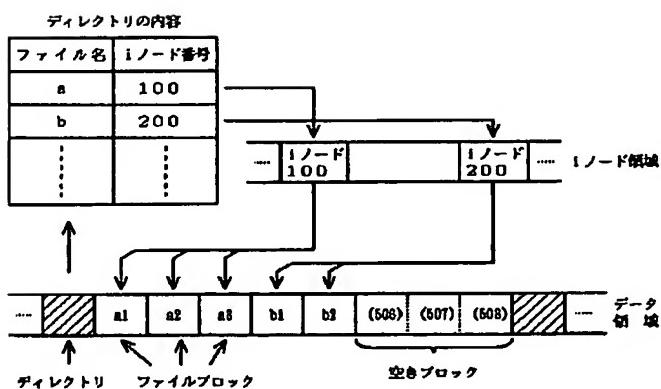
【図6】



【図3】



【図4】



【図5】

ファイル名	Iノード番号
a	100
b	200
⋮	⋮

ディレクトリ内のブロック数 (6)

ディレクトリ内の
空きブロックリスト
(508)
(507)
(506)

ディレクトリの
アクセス頻度 N

ディレクトリ移動要求フラグ

本発明のディレクトリの内容

This table, titled '本発明のディレクトリの内容' (Content of the Invention's Directory), lists files 'a' and 'b' with their I-node numbers (100 and 200). It also includes information about the number of blocks in the directory (6), a list of free blocks (508, 507, 506), access frequency (N), and a move request flag.

【図7】

